Computer Architecture

Some questions & answers

Prof. Nizamettin AYDIN, PhD

naydin@yildiz.edu.tr

http://www3.yildiz.edu.tr/~naydin

AcA-00-Introduction

- What is Computer Organization?
- What is Computer Architecture?
- Brief history of computing systems
- What are the classes of computers?
- What are the constituents of a computer?
- What are the constituents of a CPU?
- What are the constituents of a Control Unit?
- What are the levels of program code?
- What is a program?
- How do you describe The Computer Level Hierarchy?

AcA-01-Fundamentals

- What is informatics?
- What is data?
- What is information?
- What is knowledge?
- What is a system?
- What is an information system?
- What are to components that implement information system?
- What is a computing system?
- What is a digital system?
- What is a signal?

AcA-01-Fundamentals

- Compare analog and digital signals
- Why do we sample a signal?
- Why do we quantize a signal?
- Describe continuous, discrete, and digital signals.
- Describe the process of obtaining digital signals
- What is sampling theorem?
- What are the fundamental data types represented in a computing system?
- Boolean algebra, digital circuit functions

AcA-02-InstructionSet-rev

- What is an instruction?
- What is instruction set?
- What is meant by Instruction Set Architecture? Explain
- What are the general instruction types in a computing system?
- What are the elements of an instruction?
- Classify instruction set in terms of number of operands.
- What types of operand an instruction can take?
- What is Big/Little Endian?

AcA-03-Performance

- List common performance metrics used in a computing system.
- Describe the Forces on Computer Architecture.
- What type of parallelisms exist in a computing system?
- What are the classes of computers?
- What is Flynn’s Taxonomy?
- Power consumption in a processor
- How to reduce power consumption?
- What are the basic performance metrics?
- What are the measurement tools?
- What is Amdahl’ law?

AcA-04-MemoryHierarchy

- What is Memory Hierarchy?
- What is the Principle of Locality?
- What is a Cache?
- Why a Cache Memory is used?
- How many cash types exist?
- What is Main Memory
- What is Virtual Memory
- Why a Virtual Memory is used?
- Classify memory types
- Differences between SRAM and DRAM?
- Memory organization
- Virtual machines

AcA-05-Instruction-Level Parallelism

- Explain Instruction-Level Parallelism
- What is pipelining?
- What is main constraint in paralellism?
- How many dependences exist?
- What are data hazards?
- What techniques exist to avoid dependences
- What is purpose of Tomasulo’s algorithm?
- Compare the processors interms of pipelining

AcA-06-Data-Level Parallelism

- What are the classes of parallelizm? Briefly explain
- Classify computers in terms of the Data-Level Parallelism
- Brifly describe Vector Architecture
- How Vector Processors work? Explain with an example
- Brifly describe Graphics Processing Units Architecture
- What is heterogen computing system?

AcA-06-Data-Level Parallelism

- Brifly describe NVIDIA Instruction Set Architecture
- What are the Challenges for the GPU programmer
- Compare Graphics Processing Units and vector Architectures
- Dependences in Loop Level Parallelism
- How to find dependences in Loop Level Parallelism

Computer Architecture Formulas

1. *CPU time* =
   Instruction count × Clock cycles per instruction × Clock cycle time

2. X is *n* times faster than Y:
   n = Execution time$_Y$ / Execution time$_X$ = Performance$_X$ / Performance$_Y$

3. *Amdahl's Law*:
$$\text{Speedup}_{\text{overall}} = \frac{\text{Execution time}_{\text{old}}}{\text{Execution time}_{\text{new}}} = \frac{1}{(1 - \text{Fraction}_{\text{enhanced}}) + \frac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}}}$$

4. $Energy_{dynamic} \propto 1/2 \times \text{Capacitive load} \times \text{Voltage}^2$

5. $Power_{dynamic} \propto 1/2 \times \text{Capacitive load} \times \text{Voltage}^2 \times \text{Frequency switched}$

6. $Power_{static} \propto \text{Current}_{static} \times \text{Voltage}$

7. *Availability* = Mean time to fail / (Mean time to fail + Mean time to repair)

8. *Die yield* = Wafer yield × 1 / (1 + Defects per unit area × Die area)^N
   where Wafer yield accounts for wafers that are so bad they need not be tested and *N* is a parameter called the process-complexity factor, a measure of manufacturing difficulty. *N* ranges from 11.5 to 15.5 in 2011.

Computer Architecture Formulas

9. *Means—arithmetic (AM), weighted arithmetic (WAM), and geometric (GM):*

$$\text{AM} = \frac{1}{n}\sum_{i=1}^{n} \text{Time}_i\,, \qquad \text{WAM} = \sum_{i=1}^{n} \text{Weight}_i \times \text{Time}_i\,, \qquad \text{GM} = \sqrt[n]{\prod_{i=1}^{n} \text{Time}_i}$$

where Time_i is the execution time for the *i*th program of a total of *n* in the workload, Weight_i is the weighting of the *i*th program in the workload.

10. *Average memory-access time* = Hit time + Miss rate × Miss penalty

11. *Misses per instruction* = Miss rate × Memory access per instruction

12. *Cache index size*: 2^{index} = Cache size /(Block size × Set associativity)

13. $Power\ Utilization\ Effectiveness\ (PUE)$ of a Warehouse Scale Computer $= \dfrac{\text{Total Faciliy Power}}{\text{IT Equipment Power}}$

Rules of Thumb

1. Amdahl/Case Rule:
   - A balanced computer system needs about 1 MB of main memory capacity and 1 megabit per second of I/O bandwidth per MIPS of CPU performance.
2. 90/10 Locality Rule:
   - A program executes about 90% of its instructions in 10% of its code.
3. Bandwidth Rule:
   - Bandwidth grows by at least the square of the improvement in latency.

Rules of Thumb

4. 2:1 Cache Rule:
   - The miss rate of a direct-mapped cache of size *N* is about the same as a two-way set-associative cache of size *N*/2.
5. Dependability Rule:
   - Design with no single point of failure.
6. Watt-Year Rule:
   - The fully burdened cost of a Watt per year in a Warehouse Scale Computer in North America in 2011, including the cost of amortizing the power and cooling infrastructure, is about $2.

Computer Architecture

Some questions & answers

Prof. Nizamettin AYDIN, PhD
naydin@yildiz.edu.tr

http://www3.yildiz.edu.tr/~naydin

Q1

- What are the four main components of any general-purpose microprocessor?

A1

- A **memory (register)**,
  - which stores both data and instructions:
- An **arithmetic and logic unit** (ALU)
  - capable of operating on binary data;
- A **control unit**,
  - which interprets the instructions in memory and causes them to be executed;
- **Input and output (I/O) equipment**
  - operated by the control unit.

Q2

- At the integrated circuit level, what are the three principal constituents of a computer system?

A2

- Gates,
- Memory cells
- Interconnections among gates and memory cells

Q3

- List and explain the key characteristics of a computer family.

A3

- **Similar or identical instruction set**
  - In many cases, the same set of machine instructions is supported on all members of the family. Thus, a program that executes on one machine will also execute on any other.
- **Similar or identical operating system**
  - The same basic operating system is available for all family members.
- **Increasing speed**
  - The rate of instruction execution increases in going from lower to higher family members.
- **Increasing Number of I/O ports**
  - In going from lower to higher family members.
- **Increasing memory size**
  - In going from lower to higher family members.
- **Increasing cost**
  - In going from lower to higher family members.

Q4

- Consider a hypothetical 32-bit microprocessor having 32-bit instructions composed of two fields:

  The first byte contains the opcode and the remainder the immediate operand or an operand address.

  - a. What is the maximum directly addressable memory capacity (in bytes)?
  - b. How many bits are needed for the program counter and the instruction register?

A4

a. Address field of the instruction is 24 bits.

- Therefore the total memory capacity is 2^{24} = $2^4 \times 2^{20}$ = 16 Mbytes
  - (2^{20} is 1 Mega)

b. Because the address field of the instruction is 24 bits,

- The program counter must be at least 24 bits.
- If the instruction register is to contain the whole instruction, it will have to be 32-bits long

Q5

Consider a 32-bit microprocessor, with a 16-bit external data bus, driven by an 8-MHz input clock.

Assume that this microprocessor has a bus cycle whose minimum duration equals four input clock cycles.

— What is the maximum data transfer rate that this microprocessor can sustain?

A5

- Clock cycle = 1/Clock frequency
  Clock cycle = 1/8 MHz = 0.125×10^{-6} s
  $= 125 \times 10^{-9}$ s = 125 ns

- Bus cycle = number of clocks x Clock cycle
  Bus cycle = 4 x 125 ns = 500 ns

- 2 bytes transferred every 500 ns; thus

- transfer rate = $2/(500 \times 10^{-9}) = 2/(5 \times 10^{-7})$
  $= 0.4 \times 10^{7} = 4$ MBytes/sec

Q6

An 8 bit microprocessor system has 9 address lines to address relevant memory locations.

a. Assuming that the data size is 1 byte, what is the address of the last memory location?

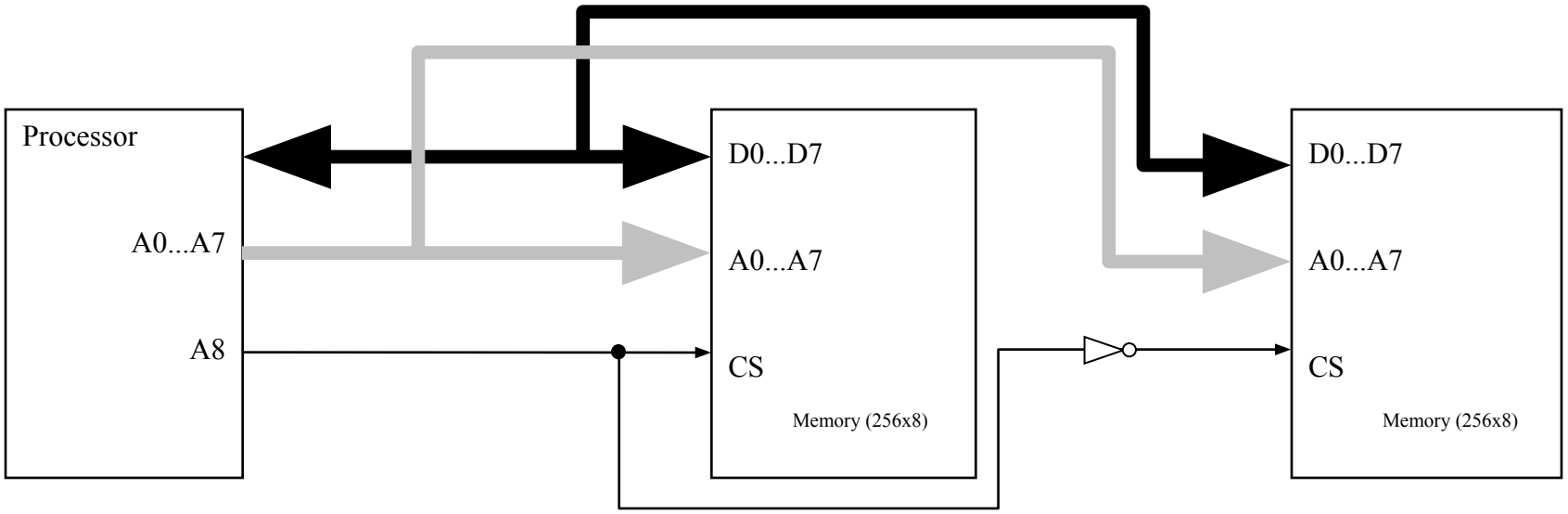
b. Design the required memory system using memory chips organized as 256x8 bits.

A6

a. Because the processor has 9 address lines the total directly addressable memory size is $2^9 = 512$ Bytes.

So, the address of the last memory location will be **$2^9-1 = 511$**

b.



Q7

A computer system has the following floating point format:

1 bit sign 5 bits biased exponent 10 bits mantissa

(bias can be taken as $2^{(\text{number of bits in exponet-1})}-1$)

- If the given number is 19.75, determine the following values:

- a. Corresponding binary number
- b. Corresponding hexadecimal number
- c. Corresponding 16 bit floating point number

A7

a. $(19.75)_{10} = (10011.11)_2$

b. $(19.75)_{10} = (0001\ 0011.1100)_2 = (13.C)_{16}$

c. $(19.75)_{10} = (10011.11)_2 => (1.001111 \times 2^{00100})$

biased exponential format= $(1.001111 \times 2^{00100+01111})$

biased exponential format = $(1.001111 \times 2^{10011})$

S = 0, BE = 10011, M = 0011110000

$(19.75)_{10} = (0\ 10011\ 0011110000)_{float}$

Q8

- A given microprocessor has words of one byte.

  What is the smallest and largest integer that can be represented in the following representation?

  a. Unsigned
  b. Sign magnitude
  c. Ones complement
  d. Twos complement
  e. Binary coded decimal

A8

The word size is 1 byte (8 bits). So,

a. 0; 255

b. –127; 127

c. –127; 127

d. –128; 127

e. 00; 99

Q9

- In a signed (2s complement number system) addition, if the "carry in" and the "carry out" of the sign bit differ, there is an overflow.

  a. Determine whether there is an overflow in the following operations or not. (use 4 bit 2s complement numbers)

     4+3; -7-6; 5+7; -3-2

  b. Design a circuit that whenever an overflow happens the output becames 1, otherwise 0.

A9.a

- If the "carry in" and the "carry out" of the sign bit differ, there is an overflow.

operation			4+3			-7-6			5+7			-3-2
Carry in			0			0			1			1
			0100			1001			0101			1101
		+	0011		+	1010		+	0111		+	1110
Result			0111			0011			1100			1011
Cary out		0			1			0			1	
overflow			no			yes			yes			no

A9.b

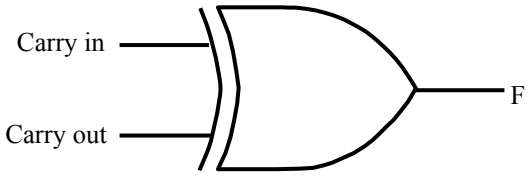
Truth table:

C_i	C_o	F
0	0	0
0	1	1
1	0	1
1	1	0

Overflow function:

$$F = C_i' C_o + C_i C_o' = C_i \oplus C_o$$

Implementation:



Computer Architecture

Some questions & answers

Prof. Nizamettin AYDIN, PhD
naydin@yildiz.edu.tr

http://www3.yildiz.edu.tr/~naydin

Q10

A computer system has an 8 bit accumulator and 12 bit address bus.

If the content of memory location 1F0 is the binary number 11000011;

(a) what is the corresponding decimal value when the number represents an unsigned integer number?

(b) what is the corresponding decimal value when the number represents a twos complement number?

(c) what is the corresponding decimal value when the number represents a ones complement number?

(d) Assuming that the most significant bit is parity bit, what is the corresponding letter when the number represents an ASCII character (hint: ASCII character is represented by 7 bits and 41H represents A)?

A10

Binary 11000011:

(a) $1\times2^7+1\times2^6+1\times2^1+1\times2^0 = 195$

(b) $-1\times2^7+1\times2^6+1\times2^1+1\times2^0 = -61$

(c) -60

(d) If the most significant bit is parity bit, ASCII character represented by 11000011 is 100 0011 (43 in hexadecimal) corresponding to capital letter C.

Q11

Consider a dynamic RAM that must be given a refresh cycle 64 times per ms.

- Each refresh operation requires 150 ns;

What percentage of the memory's total operating time must be given to refreshes?

A11

- In 1 ms, the time devoted to refresh is:
  64 × 150 ns = 9600 ns = 9.6 μs
  =0.0096 ms.

- The fraction of time devoted to memory refresh is:
  $(9.6 \times 10^{-6}\ \text{s})/10^{-3}\ \text{s} = 0.0096$,
  which is 0.96%.

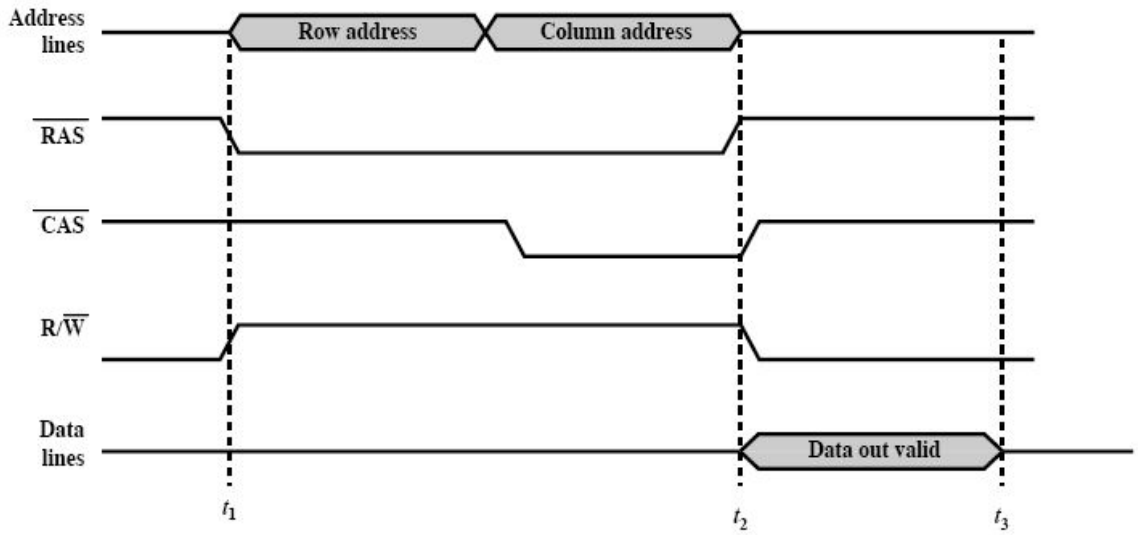
Q12

- Following figure shows a simplified timing diagram for a DRAM read operation over a bus.
- The access time is considered to last from t_1 to t_2. Then there is a recharge time, lasting from t_2 to t_3, during which DRAM chips will have to recharge before the processor can access them again.

a. Assume that the access time is 60 ns and the recharge time is 40 ns. What is the memory cycle time?

b. What is the maximum data rate this DRAM can sustain, assuming a 1-bit output?

c. Constructing a 32-bit wide memory system using these chips yields what data transfer rate in terms of Byte per second (B/s)?



A12

a. Memory cycle time = 60 + 40 = 100 ns.

b. The maximum data rate is 1 bit every 100 ns, which is 10 Mb/s.

c. 32×10 Mbps = 320 Mbps
= 320 Mbps / 8 = 40 MB/s.

Q13

In the Intel 8088 microprocessor, a bus read operation requires four processor clock cycles.

Assuming a processor clock rate of 8 MHz (fc = 8 Mega Hertz);

a. what is the maximum data transfer rate? (Data transfer rate is the total number of bytes transferred in one second)

b. what is the maximum data transfer rate if a one clock cycle wait state is inserted per byte transferred? (In wait state processor does nothing)

A13

a. The clock period is Tcc = 1/fc = 1/8 MHz
=125 ns = 125×10^{-9} s

One bus read cycle is Tbc = 4×Tcc = 4×125
=500 ns = 0.5 μs.

If the bus cycles repeat one after another;

maximum data transfer rate is $1/(0.5\times10^{-6})$ = 2×10^{6} Byte per second = 2 MB/s.

b. The wait state extends the bus read cycle by 125 ns, for a total duration of 0.625 μs.

The corresponding data transfer rate is 1/0.625 = 1.6 MB/s.

Q14

- If one line of an assembler program (which loads contents of memory location 940 to the accumulator) for a hypothetical processor and corresponding machine code is given as:

  address assembler machine code
  300 lda 940 1940 (the values are in hexd.)

a. How many bits are needed for the program counter?

b. How many bits are needed for the instruction register?

c. What is the maximum directly addressable memory capacity in KBytes?

d. For the memory location 940, find the Tag, Line, and Word values in hexadecimal format for a direct-mapped cache, when tag-id=4 bits, line-id=6 bits, word-id=2 bits.

A14

a. Because the address size is 3 nibbles (3x4=12 bits), a 12 bit Program Counter is required.

b. Instruction register size equals to *op-code size* + *address size*(4+12), which is 16 bits.

c. Maximum memory capacity = 2^{12} bytes = 2^{10+2} bytes = $2^2 \times 2^{10}$ bytes = 4 Kbytes

d. First, write the address in binary format:
$(940)_{16} = (1001\ 0100\ 0000)_2$
Then define the Tag, Line, and Word values:

Tag-id← 100101000000 →Word-id
Line-id

Tag-id = 1001 = 9
Line-id = 00010000 = 10
Word-id = 0000 = 0

Q15

Given the hexadecimal main memory address A1F85B73;

a. How many bits are needed for the program counter?

b. How many bits are needed for the instruction register if op-code is 8 bits.

c. What is the maximum directly addressable memory capacity in GByte?

d. Find the Tag, Line, and Word values in hexadecimal format for a direct-mapped cache, when *tag-id*=13 bits, *line-id*=16 bits, *word-id*=3 bits.

A15

a. Because the address size is 4 bytes (32 bits), a 32 bit Program Counter is required.

b. Instruction register size equals to *op-code size* + *address size*, which is 40 bits.

c. Maximum memory capacity = 2^{32} bytes = 2^{30+2} bytes = $2^2 \times 2^{30}$ bytes = 4 Gbytes

d. First, write the address in binary format: 1010 0001 1111 1000 0101 1011 0111 0011

Then define the Tag, Line, and Word values:

Tag-id← 10100001111110000101101101110011 →Word-id

Line-id

Tag-id = 0001010000111111 = 143F

Line-id = 0000101101101110 = 0B6E

Word-id = 0011 = 3

Q16

Consider a machine with a byte addressable main memory of 2^{16} bytes and block size of 8 bytes.

Assume that a direct mapped cache consisting of 32 lines is used with this machine.

a. How is the memory address divided into tag, line number, and byte number?

b. Into what line would bytes with each of the following addresses be stored?

i. 0001 0001 0001 1011

ii. 1100 0011 0011 0100

A16

a. Address size is 16 bits word-id= $\log_2$(block size) = $\log_2$(8) = 3 bits line-id = $\log_2$(#of lines) = $\log_2$(32) = 5 bits tag-id = 16 – (5+3) = 8 bits

b. i. 00010001 00011 011 ☐ line id = 03

ii. 11000011 00110 100 ☐ line id = 06

Computer Architecture

Some questions & answers

Prof. Nizamettin AYDIN, PhD
naydin@yildiz.edu.tr

http://www3.yildiz.edu.tr/~naydin

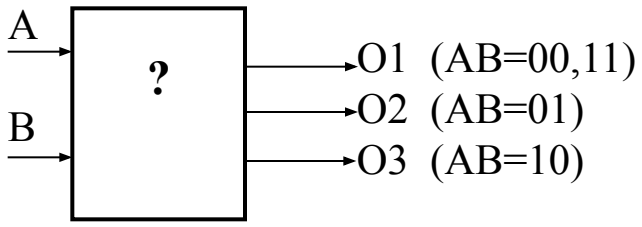
Q17

In an algorithm implemented in hardware, two bits (say A and B) are checked at each step to determine one of the three operations:

- Operation 1, if (A=0, B=0) or (A=1, B=1)
- Operation 2, if (A=0, B=1)
- Operation 3, if (A=1, B=0)

Design the required logic circuit.

A17

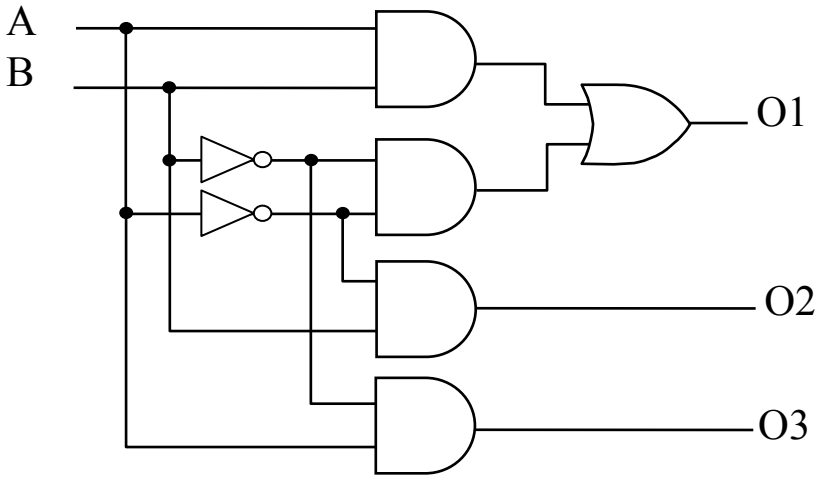


A	B	O1	O2	O3
0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
1	1	1	0	0

$$O1 = \overline{A}\,\overline{B} + A\,B$$

$$O2 = \overline{A}\,B$$

$$O3 = A\,\overline{B}$$



Q18

X is a decimal number and its value is 51.

(a) Represent X in unsigned binary integer form using 8 bits,

(b) Find the corresponding hexadecimal representation of X,

(c) Assuming X is a 2s complement signed integer number using 8 bits, find –X.

(d) Represent X in BCD

A18

(a) $X = (51)_{10} = (00110011)_2$

(b) $X = (51)_{10} = (0011\ 0011)_2 = (33)_{16}$

(c) $X = (51)_{10} = (00110011)_2$ □ $-X = -(51)_{10}$
= 2s complement of $(X)_2 = (11001101)_2$

(d) $X = (51)_{10} = (0101\ 0001)_{BCD}$

Q19

The following table shows a memory in a computer system.

(a) How many memory locations this memory has?

(b) What is the memory size (in Bytes)?

(c) Total how many bits this memory can store?

Memory location	Memory Content
000000	1010110010101100
000001	0010110110101100
. . .	. . .
111110	1010110010111101
111111	0000110110101100

A19

(a) Because the address field of this memory is 6 bits, there are 2^6 = 64 memory locations.

(b) Each location stores 2 Byte (16 Bits) data. So the memory size is number of locations × size of data at each location = 64×2 = 128 Byte.

(c) Because a byte is 8 bits, this memory can store maximum 128×8 = 1024 bits data.

Q20

If one line of an assembler program for a hypothetical processor, which uses bytewise addressing, and corresponding machine code is given as:

address	assembler	machine code
2000	add F1FA, B1B1	12F1FAB1B1

(a) How many instructions this processor might have?

(b) How many bits are needed for the instruction register?

(c) How many memory location an instruction will occupy?

(d) What is the total directly addressable memory size ?

A20

address	assembler	machine code
2000	add F1FA, B1B1	12F1FAB1B1

(a) Because the op-code field of this instruction is 2 hexadecimal digits (8 bits) there might be maximum $2^8 = 256$ instructions

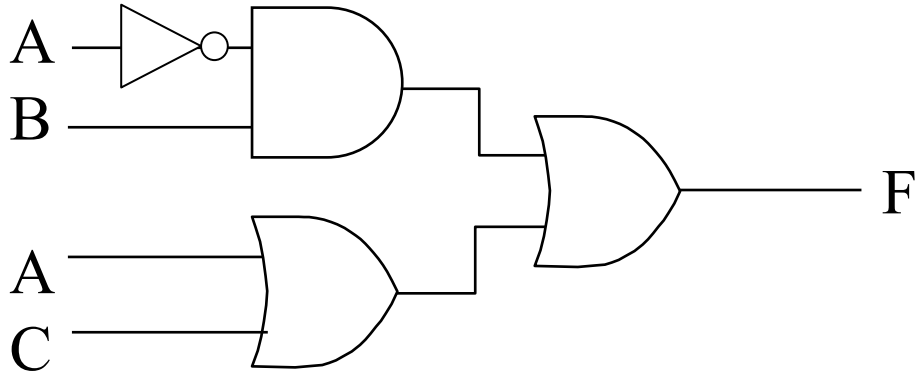
(b) Because the instruction length is 10 hexadecimal digits, at least $4 \times 10 = 40$ bits are needed for the instruction register.

(c) Each location stores 1 Byte (8 Bits) data. The instruction length is 5 bytes. Therefore each instruction will occupy 5 memory locations.

(d) Address field of this instruction has 4 hexadecimal digits, so the total memory size is $2^{4\times4} = 2^{16}$ Byte $= 2^{10+6}$ Byte $= 2^6 \times 2^{10}$ Byte = 64 kByte.

Q21

Given the following network;



(a) Write an expression for the network,

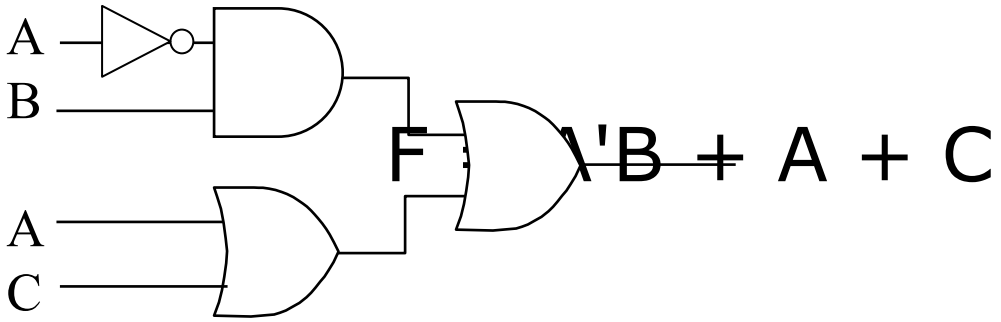
(b) Simplify the expression.

(c) Write out a truth table for the expression.

(d) Draw the network of the simplified expression.

A21

(a)



(b)

F = A'B + A + C

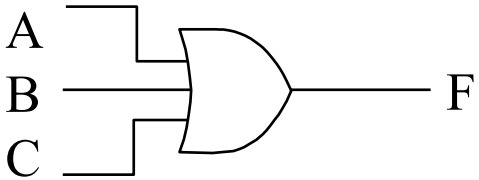
F = (A + A')(A + B) + C

F = A + B + C

(c) Truth table:

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

(d)



Q22

Convert the given decimal number $(-18.75)_{10}$ to the corresponding 32 bit IEEE floating point number.

A22

- Step 1. Convert the number to binary

  $(18.75)_{10} = (10010.11)_2$

- Step 2. Normalize the binary number to 1

  $(10010.11)_2 = (1.001011 \times 2^4) = (1.001011 \times 2^{00000100})$

- Step 3. Add bias (127) to exponent to obtain biased exponential format

  biased exponential format = $(1.001011 \times 2^{4+127}) = (1.001011 \times 2^{00000100+01111111}) = (1.001011 \times 2^{10000011})$

  S = 1, BE = 10000011, M = 00101100000000000000000

- Step 4. Join S, BE and M to form binary FP number

  $(-18.75)_{10} = (1\ 10000011\ 00101100000000000000000)_{fp}$

  $(-18.75)_{10} = (C1960000)_{\text{fp in hexadecimal}}$

Q23

Write a simple VVM Assembly program that adds *the number entered from the keyboard* and *the data in memory location **20***.

If the result is positive, it is saved in memory location **30**;

If the result is negative, it is saved in memory location **40**

A23

One solution is as follows:

```
00 in       // input number
01 add 20   // accumulator + content of (20)
02 brp 05   // if +ve jump to 05
03 sto 40   // else store –ve number in (30)
04 jmp 06   // jump (branch) to 06
05 sto 30   // save the +ve number in (30)
06 hlt      // stop
```

Q24

- Consider an array of five drives (X0, X1, X2, X3 contain data, X4 is parity disk).
- Parity of *i*th bit is calculated as $X4(i) = X3(i) \oplus X2(i) \oplus X1(i) \oplus X0(i)$
- Suppose that drive **X2** has failed.
- Show how to regenerate the contents of **X2.** ?

A24

- The contents of X2 can be regenerated by XORing both sides by X4 and X2 :

$$X2(i) \oplus X4(i) \oplus X4(i) = X3(i) \oplus X2(i) \oplus X1(i) \oplus X0(i) \oplus X2(i) \oplus X4(i)$$

- Because $X4(i) \oplus X4(i) = 0$ and $X2(i) \oplus X2(i) = 0$;

$$X2(i) = X4(i) \oplus X3(i) \oplus X1(i) \oplus X0(i)$$

Computer Architecture

Some questions & answers

Prof. Nizamettin AYDIN, PhD
naydin@yildiz.edu.tr

http://www3.yildiz.edu.tr/~naydin

Q25

- List three broad classifications of external (or peripheral) devices.

A25

- **Human readable**:
  - Suitable for communicating with the computer user.
- **Machine readable**:
  - Suitable for communicating with equipment.
- **Communication**:
  - Suitable for communicating with remote devices

Q26

- Given x = 0101 and y = 1010 in 2s complement notation (i.e., x = 5, y = -6), compute the product p = x × y with Booth's algorithm.

A26

A	Q	M		
Initialization:	$Q_3Q_2Q_1Q_0$		Q_{-1}	
0000	1010	0	0101	Initial values
1st cycle:				
0000	0101	0	0101	AShiftr
2nd cycle:				
1011	0101	0	0101	A□A-M
1101	1010	1	0101	AShiftr
3rd cycle:				
0010	1010	1	0101	A□A+M
0001	0101	0	0101	AShiftr
4th cycle:				
1100	0101	0	0101	A□A-M
1110	0010	1	0101	AShiftr

Result is in A and Q

Q27

- Given x = 1001 and y = 0010 in twos complement notation (i.e., x = -7, y = 3), compute the division p = x / y .

A27

Accumulator	Quotient	Mdivisor	
$A_3A_2A_1A_0$	$Q_3Q_2Q_1Q_0$	$M_3M_2M_1M_0$	Divident is in A and Q
1111	1001	0011	Initial values
1st cycle:			
1111	0010	0011	LShiftl
0010	0010	0011	A□A+M (if $A_3 \neq M_3$)
1111	0010	0011	Restore A, Q_0□0 (if A ≠ 0)
2nd cycle:			
1110	0100	0011	LShiftl
0001	0100	0011	A□A+M (if $A_3 \neq M_3$)
1110	0100	0011	Restore A, Q_0□0 (if A ≠ 0)
3rd cycle:			
1100	1000	0011	LShiftl
1111	1000	0011	A□A+M (if $A_3 \neq M_3$)
1111	1001	0011	Q_0□1 (if $A_{3a} = A_{3b}$)
4th cycle:			
1111	0010	0011	LShiftl
0010	0010	0011	A□A+M (if $A_3 \neq M_3$)
1111	0010	0011	Restore A, Q_0□0 (if A ≠ 0)

Remainder is in A and quotient in Q

Q28

- In a computer system, address 100 contains decimal value 32, address 200 contains decimal value 10.
- What would be the contents of accumulator after running the following assembler code.
- Explain what happens.
  - LOAD 100
  - SHIFTR
  - SHIFTR
  - ADD 200

A28

- If address 100 contains 32, address 200 contains 10:

Instruction	*Acc. Content*	*Operation*
LOAD 100	A=32	A ☐ M(100)
SHIFTR	A=16	A ☐ A/2
SHIFTR	A=8	A ☐ A/2
ADD 200	A=18	A ☐ A+M(200)

Q29

- In a computer system, a small part of memory is given in the following table. What would be the contents of accumulator after running the following assembler code. (All values are in hexadecimal).

Mem. Adress	Data
A0	A4
A1	A3
A2	22
A3	3A
A4	A1

- LOAD IMMEDIATE A1
- RROTATE
- ADD INDIRECT A4
- AND IMMEDIATE EA
- SUB DIRECT A2
- SHIFTL

A29

LOAD IMMEDIATE A1 $\quad Acc = (1010\ 0001)_2 = (A1)_{16}$

RROTATE $\quad Acc = (1101\ 0000)_2 = (D0)_{16}$

ADD INDIRECT A4 $\quad Acc = (1101\ 0000 + 1010\ 0011)_2$
$= (0111\ 0011)_2 = (73)_{16}$

AND IMMEDIATE EA $\quad Acc = (0111\ 0011\ AND\ 1110\ 1010)_2$
$= (0110\ 0010)_2 = (62)_{16}$

SUB DIRECT A2 $\quad Acc = (0110\ 0010 - 0010\ 0010)_2$
$= (0100\ 0000)_2 = (40)_{16}$

SHIFTL $\quad Acc = (1000\ 0000)_2 = (80)_{16}$

Q30

Given the following memory values and a one-address machine with an accumulator, what values do the following instructions load into the accumulator?

Word 20 contains 40;
Word 30 contains 50;
Word 40 contains 60;
Word 50 contains 70;

a. LOAD IMMEDIATE 20
b. LOAD DIRECT 20
c. LOAD INDIRECT 20
d. LOAD IMMEDIATE 30
e. LOAD DIRECT 30

A30

Word 20 contains 40; Word 30 contains 50; Word 40 contains 60; Word 50 contains 70;

a. LOAD IMMEDIATE 20
b. LOAD DIRECT 20
c. LOAD INDIRECT 20
d. LOAD IMMEDIATE 30
e. LOAD DIRECT 30

a. *20*

b. *40*

c. *60*

d. *30*

e. *50*

Q31

- If the last operation performed on a computer with an 8 bit word was an addition in which the two operands were 2 and 3, what would be the value of the following flags:
  - Carry flag
  - Zero flag
  - Overflow flag
  - Sign flag
- What if the operands were -1 (2's complement) and +1?

A31a

2 (8 bit) 00000010
3 (8 bit) 00000011
00000101

Carry = 0

Zero = 0

Overflow = 0

Sign = 0

A31b

-1 (8 bit 2s Complement) 11111111
1 (8 bit 2s Complement) 00000001
1 00000000

Carry = 1

Zero = 1

Overflow = 0

Sign = 0

Q32

- Let the address stored in the program counter be designated by the symbol X1.
- The instruction stored in X1 has an address part (operand reference) X2. The operand needed to execute the instruction is stored in the memory word with addres X3.
- An index register contains the value X4.
- What is the relationship between these various quantities if the addressing mode of instruction is

  a. direct,

  b. indirect,

  c. indexed,

  d. PC relative?

A32

a. X3 = X2

b. X3 = (X2)

c. X3 = X2 + X4

d. X3 = X1 + X2 + 1

Q33

A PC-relative mode branch instruction is 3 bytes long. The address of instruction, in decimal, is 256028. Determine the branch target address if the signed displacement in instruction is -31.

A33

Recall that relative addressing uses the contents of the program counter, which points to the next instruction after the current instruction.

In this case, the current instruction is at decimal address 256028 and is 3 bytes long, so the PC contains 256031.

With the displacement of –31, the effective address is 256000.

Computer Architecture

Some questions & answers

Prof. Nizamettin AYDIN, PhD
naydin@yildiz.edu.tr

http://www3.yildiz.edu.tr/~naydin

Q34

- What is an instruction and instruction set? Explain.

A34

- Instruction:
  - Language of the machine
- Instruction set:
  - Vocabulary of the language (collection of instructions that are understood by a CPU)

Q35

- Consider a stack-based processor with instructions

  PUSH, POP, ADD, SUB , MUL, and DIV.

- Write a program to compute

  $X = (A + B^2)/(D^2 - E)$

A35

- $X = (A + B^2)/(D^2 - E)$

```
PUSH A
PUSH B
PUSH B
MUL
ADD
PUSH D
PUSH D
MUL
PUSH E
SUB
DIV
POP X
```

Q36

A one-address type processor has an 8 bit accumulator (A), and a bytewise memory addressing. Data in memory locations 50 and 60 are 7A and 55 respectively, and they are in twos complement format. The following code is a part of an assembler program. If the code is executed;

```
I0    LDA (50)
I1    ADD (60)
I2    BRN I7
I3    LDA (51)
.....
.....
I7    STA (70)
I8    JMP I3
```

a. What will be the value of overflow flag after the execution of instruction I1? Explain
b. What will be the value of carry flag after the execution of instruction I1? Explain
c. Which line will be executed after the execution of instruction I2? Explain
d. What will be the content of program counter (PC)after the execution of I8.

LDA (X) : Load contents of memory location X to accumulator,
ADD (X) : Add contents of memory location X to accumulator,
BRN X : Branch to line X if negative,
STA : Store contents of accumulator to memory location X,
JMP X : Unconditional jump to line X

A36

a. Overflow flag is 1.

- Both 7A and 55 are positive. But the result is negative:

$$\begin{array}{rl} 7A & (01111010) \\ +\,55 & (01010101) \\ \hline CF & (11001111) \end{array}$$

b. Carry flag will be 0, as there is no carry.

c. Because the sign flag is 1 (the result is negative), instruction I7 will be executed after the execution of the instruction I2

d. The program counter content will be I3

Q37

The contents of memory and CPU registers of a computer system is given as following (the values are in **hexadecimal**)**.**

What will be the actual operand of the instruction for the;

- immediate addressing
- direct addressing
- indirect addressing
- PC relative addressing

	Memory		CPU registers	
100	**1240**		**101**	**PC**
101	**5241**		**0003**	**AC**
102	**2241**		**5241**	**IR**
	.			
240	**0003**			
241	**0343**			
	.			
342	**0003**			
343	**0302**			

A37

a. 0241

b. 0343

c. 0302

d. 0003

Q38

- Assume a stack-based processor that includes the stack operations PUSH and POP. Arithmetic operations automatically involve the top one or two stack elements. Begin with an empty stack. What stack elements remain after the following instructions are executed?

```
PUSH 4
PUSH 7
PUSH 8
ADD
PUSH 10
SUB
MUL
```

A38

Instruction	Stack (top on the left)
PUSH 4	4
PUSH 7	7, 4
PUSH 8	8, 7, 4
ADD	(8+7=15), 4
PUSH 10	10, 15, 4
SUB	(15-10=5), 4
MUL	(5×4=20)

Q39

- Compare zero-, one-, two-, and three-address machines by writing programs to compute

  $$X = (A + B \times C)/(D - E \times F)$$

  for each of the four machines (0 Address, 1 Address, 2 Address, 3 Address machines).

  The instructions available for use are as follows:

0 Address	1 Address	2 Address	3 Address
PUSH M	LOAD M	MOV (X ☐ Y)	MOV (X ☐ Y)
POP M	STORE M	ADD (X ☐ X + Y)	ADD (X ☐ Y + Z)
ADD	ADD M	SUB (X ☐ X - Y)	SUB (X ☐ Y - Z)
SUB	SUB M	MUL (X ☐ X × Y)	MUL (X ☐ Y × Z)
MUL	MUL M	DIV (X ☐ X / Y)	DIV (X ☐ Y / Z)
DIV	DIV M		

A39

$$X = (A + B \times C)/(D - E \times F)$$

0 Address
(Stack Machines)

```
PUSH A
PUSH B
PUSH C
MUL
ADD
PUSH D
PUSH E
PUSH F
MUL
SUB
DIV
POP X
```

1 Address
(Accumulator Machine)

```
LOAD E
MUL F
STORE T
LOAD D
SUB T
STORE T
LOAD B
MUL C
ADD A
DIV T
STORE X
```

2 Address
(Memory-Memory, or Register-register, or Memory-register Machine)

```
MOV R0, E
MUL R0, F
MOV R1, D
SUB R1, R0
MOV R0, B
MUL R0, C
ADD R0, A
DIV R0, R1
MOV X, R0
```

3 Address
Load-Store

```
MUL R0, E, F
SUB R0, D, R0
MUL R1, B, C
ADD R1, A, R1
DIV X, R0, R1
```

Q40

- Assume a pipeline with 4 stages:
  - fetch instruction (FI),
  - decode instruction and calculate addresses (DA),
  - fetch operand (FO), and
  - execute (EX).

- Draw the pipelinening diagram for a sequence of 7 instructions, in which the third instruction is a branch to instruction 15 that is taken and in which there are no dependencies.

A40...

	1	2	3	4	5	6	7	8	9	10
I1										
I2										
I3										
I4										
I5										
I6										
I15										

...A40

	1	2	3	4	5	6	7	8	9	10
I1	FI	DA	FO	EX						
I2		FI	DA	FO	EX					
I3			FI	DA	FO	EX				
I4				FI	DA	FO				
I5					FI	DA				
I6						FI				
I15							FI	DA	FO	EX

Q41

- A pipelined processor has a clock rate of 2.5 GHz and executes a program with 1.5 million instructions.
  - The pipeline has five stages, and instructions are issued at a rate of one per clock cycle.
  - Ignore penalties due to branch instructions and out-of-sequence executions.

  What is the speedup of this processor for this program compared to a nonpipelined processor?

A41

We can ignore the initial filling up of the pipeline and the final emptying of the pipeline, because this involves only a few instructions out of 1.5 million instructions.

Therefore the speedup is a factor of five.

Q42

- Provide a typical list of the inputs and outputs of a control unit.

A42

- **Inputs** :
  - **Clock:**
    - This is how the control unit "keeps time." The control unit causes one micro-operation (or a set of simultaneous micro-operations) to be performed for each clock pulse. This is sometimes referred to as the processor cycle time, or the clock cycle time.
  - **Instruction register:**
    - The opcode of the current instruction is used to determine which micro-operations to perform during the execute cycle.
  - **Flags:**
    - These are needed by the control unit to determine the status of the processor and the outcome of previous ALU operations.
  - **Control signals from control bus:**
    - The control bus portion of the system bus provides signals to the control unit, such as interrupt signals and acknowledgments. The
- **Outputs** :
  - **Control signals within the processor:**
    - These are two types: those that cause data to be moved from one register to another, and those that activate specific ALU functions.
  - **Control signals to control bus:**
    - These are also of two types: control signals to memory, and control signals to the I/O modules.

Q43

- Your ALU can add its two input registers, and it can logically complement the bits of either input register, but it cannot subtract.
- Numbers are to be stored in two's complement representation.

- List the micro-operations your control unit must perform to cause a subtraction.

A43

- Consider the instruction SUB R1, X, which subtracts the contents of location X from the contents of register R1, and places the result in R1.

- t1: MAR ☐ (IR(address))
- t2: MBR ☐ Memory
- t3: MBR ☐ Complement(MBR)
- t4: MBR ☐ Increment(MBR)
- t5: R1 ☐ (R1) + (MBR)

Computer Architecture

Some questions & answers

Prof. Nizamettin AYDIN, PhD
naydin@yildiz.edu.tr

http://www3.yildiz.edu.tr/~naydin

Q44

- In the following MIPS code.

```
lw  $t0,    32($s3)
add    $t0,    $s2,    $t0
sw $t0,    48($s3)
```

a. How many times is instruction memory accessed?

b. How many times is data memory accessed?

c. How many times is register file accessed?

d. Which ones are read from the register file, which ones are write to the register file?

A44

a. 3

b. 2

c. 7

d.

```
lw    $t0, 32($s3) □ s0 is read from,
                     t0 is written into
add   $t0, $s2, $t0 □ s2 and  t0 are read from,
                     t0 is written into
sw    $t0, 48($s3) □ s3 and t0 are read from
```

Q45

- Suppose a system has a virtual memory size of 64K and the system uses word addressing.

- If the page size is 1024 Word,

  a. How many bits this virtual memory address requires?

  b. Determine the number of virtual pages.

A45

- Virtual address space:
  - $64K = 2^6 \times 2^{10} = 2^{16}$ Word
- Page size:
  - $1024 \text{ Word} = 2^{10}$ Word

a. Because the virtual memory size is 2^{16} Word, the virtual memory address requires:
- **16 bits**

b. Number of virtual pages:
- $2^{16} / 2^{10} = 2^{16-10} = 2^6 =$ **64**

Q46

- Assume that you have a computer with a maximum main memory size of 2 GByte
  - (motherboard does not allow to upgrade the main memory).
- The program you want to run on this computer may require a larger main memory, resulting in a virtual memory that uses part of the hard drive.
  - However, you realized that the magnetic hard drive in this computer is too slow for your application.
- What can you do to increase the performance of this computer?
  - Explain how and why your solution will increase the performance.

A46

- I would replace magnetic hard drive with a Solid State Drive (SSD).

- SSDs have the following advantages over HDDs:
  - High-performance input/output operations per second (IOPS),
  - Less susceptible to physical shock and vibration
  - Longer lifespan
  - Lower power consumption
  - Lower access times and latency rates
  - Silent operation

Q47

- Consider a processor which uses branch delay technique for optimization of pipelining.

```
100    LOAD    R1, (500)
101    JUMP    105
102    ADD     R1, 1
103    AND     R1, (501)
```

- Each intruction has 2 or 3 stages depending on the instruction type
  - I: Instruction fetch, E: Execute
    D: Memory operation

a. Which instruction will be executed first after the processor encounters the instruction in line 101.

Q47

b. Complete the following pipelinening diagram for the given program.

- I: Instruction fetch, E: Execute
- D: Memory operation

```
100  LOAD   R1, (500)
101  JUMP   105
102  ADD    R1, 1
103  AND    R1, (501)
```

	1	2	3	4	5	6
100 LOAD R1, (500)	I	E	D			
101 JUMP 105		I	E			
102 ADD R1, 1			I	E		
103 AND R1, (501)				I	E	D

A47

a. First, instruction in line 100, which adds contents of memory location 500 to R1 will be executed.

- In line 101 there is a jump (unconditional branch) instruction.
- Because the processor uses a branch delay technique, instruction in line 102 will be executed before processor jumps to line 105.

b.

	1	2	3	4	5	6
100 LOAD R1, (500)	I	E	D			
101 JUMP 105		I	E			
102 ADD R1, 1			I	E		
103 AND R1, (501)				I	E	D

Computer Architecture

Some questions & answers

Prof. Nizamettin AYDIN, PhD

naydin@yildiz.edu.tr

http://www3.yildiz.edu.tr/~naydin

Q48 – A48

- Suppose that a given process requires a virtual address space of 2^8 words and physical addresses in the computing system contain 7 bits.
- Assume also that pages are 32 words in length

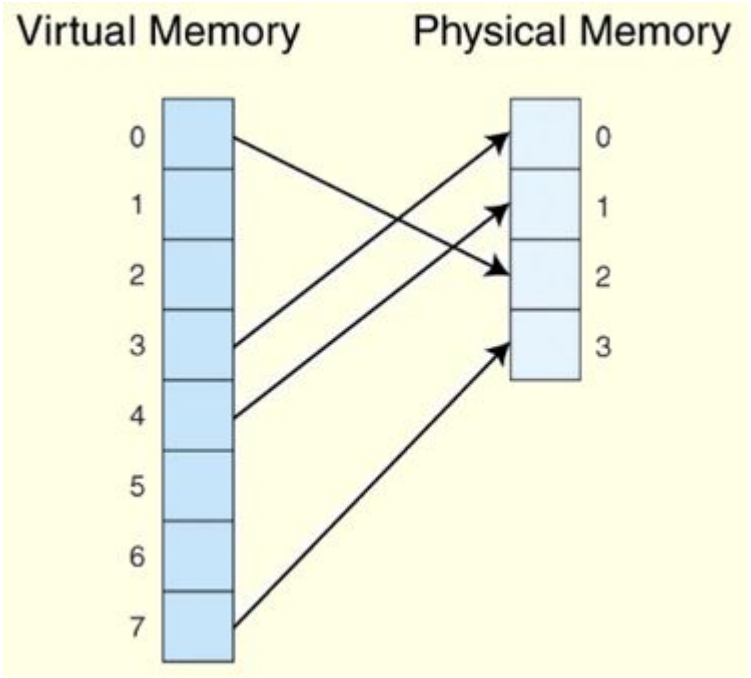
a. How many frames does the physical memory has?
  - 2^7 / 32 = 128 / 32 = 4 frames

b. How many pages does the virtual memory has?
  - 2^8 / 32 = 256 / 32 = 8 pages

Q48 – A48

- If some pages from the process have been brought into main memory as illustrated in the following figure,

c. what will be the contents of the page table?



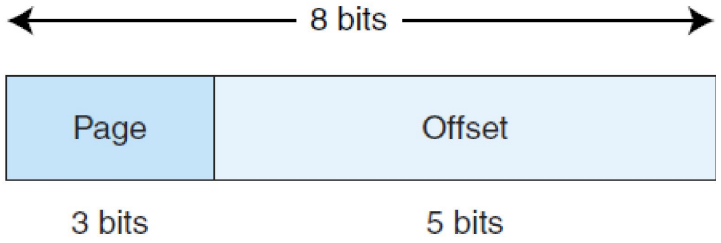
Page Page Table

Page	Frame #	Valid Bit
0	2	1
1	-	0
2	-	0
3	0	1
4	1	1
5	-	0
6	-	0
7	3	1

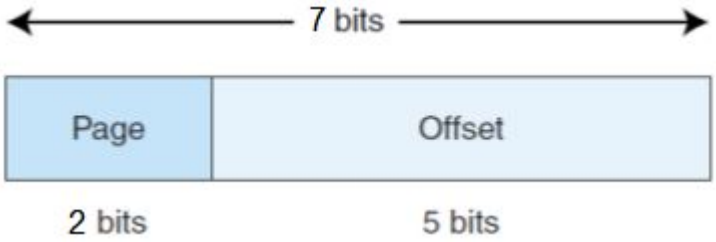
Q48-A48

- The logical page number is translated into a physical page frame through a lookup in the page table.

d. What will be the format for the virtual address?



e. What will be the format for the physical address?

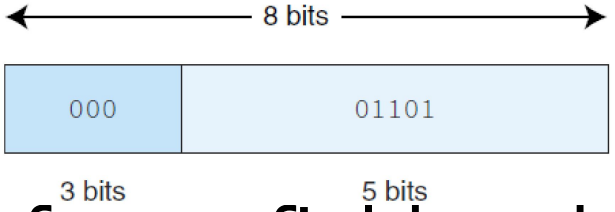


Q48-A48

- Suppose the system now generates the virtual address $13_{10} = 0D_{16} = 00001101_2$.

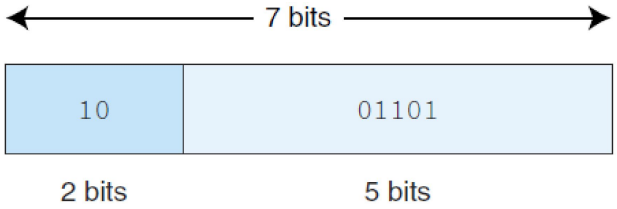
f. What will be the values of page field and offset field of virtual address?

- the page field = 0 = 000_2
- the offset field = 13 = 01101_2



g. What will be the values of frame field and offset field of physical address?

- the frame field = 2 = 10_2
- the offset field = 13 = 01101_2



Q49

- The following code is a part of an assembler program written for a three address machine type processor.

```
I1: ADD   R7, R5, R6
I2: MUL   R8, R7, R4
I3: ADD   R4, R2, R1
I4: DIV   R8, R1, R3
```

- Identify the type of all possible dependencies in this code.
  - true data or write-read dependency
  - output or write-write dependency
  - antidependency or read-write dependency

A49

- True data or write-read dependency : I1, I2 (R7, R7)

  I1: **ADD R7, R5, R6**
  I2: **MUL R8, R7, R4**
  I3: **ADD R4, R2, R1**
  I4: **DIV R8, R1, R3**

- Output or write-write dependency : I2, I4 (R8, R8)

  I1: **ADD R7, R5, R6**
  I2: **MUL R8, R7, R4**
  I3: **ADD R4, R2, R1**
  I4: **DIV R8, R1, R3**

- Antidependency or read-write dependency : I2, I3 (R4, R4)

  I1: **ADD R7, R5, R6**
  I2: **MUL R8, R7, R4**
  I3: **ADD R4, R2, R1**
  I4: **DIV R8, R1, R3**

Computer Architecture

Some questions & answers

Prof. Nizamettin AYDIN, PhD

naydin@yildiz.edu.tr

http://www3.yildiz.edu.tr/~naydin

Q50

- MIPS processor was born in the early 1980s from the work done by John Hennessy and his students at Stanford University and is mainly used in small devices. So,

  - What architecture the MIPS processor adapted?
  - What are the advantages of this architecture?

A50

- MIPS adapted the RISC architecture to explore the architectural concept of RISC

- Advantages of RISC Architecture:
  - The performance of RISC processors is often two to four times than that of CISC processors because of simplified instruction set.
  - This architecture uses less chip space due to reduced instruction set.
  - The per-chip cost is reduced by this architecture that uses smaller chips consisting of more components on a single silicon wafer.
  - RISC processors can be designed more quickly than CISC processors due to its simple architecture.
  - The execution of instructions in RISC processors is high due to the use of many registers for holding and passing the instructions as compared to CISC processors.

Q51

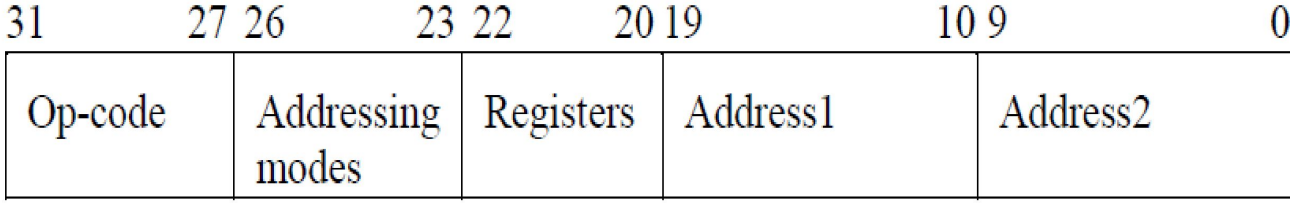
- List the characteristics of the Von Neumann computer model.

A51

- The Von Neumann computer model has the following characteristics:
  - Three hardware systems:
    - A central processing unit (CPU)
    - A main memory system
    - An I/O system
  - The capacity to carry out sequential instruction processing.
  - A single data path between the CPU and main memory.

Q52

- If a computer system, in which addressing is by 1 byte unit, has the following instruction format:



  - How many registers this computer system might have?
  - How many bits are needed for the program counter?
  - How many memory locations this instruction will occupy?
  - Total how many bits can be stored in the memory of this computer?

A52

- 2^3 = 8 registers,
- 10 bits are needed
- 32 / 8 = 4 memory locations
- 2^{10} = 1024 Byte = 1024×8 = 8192 bit,

Q53

- Data from a damaged file containing variables in 32 bit IEEE floating point number format needs to be recovered. One of the recovered data appears to be $(432AA000)_{fp(hex)}$.
  - What is the recovered data in decimal?
  - Find corresponding BCD code of the recovered data.

A53

- $(X_{10}=(-1)^S \times 1.M \times 2^{BE-127})$

  $(432AA000)_{fp(hex} = (01000011001010101010000000000000)_{fp(bin)}$

  $S = 0$

  $BE = (134)_{10} = (10000110)_2$ □ $E = (134\text{-}127)_{10} = (10000110 - 01111111)_2 = (00000111)_2$

  $M = 01010101010000000000000$

  $X_{10}=(-1)^S \times 1.M \times 2^{BE-127}) = (-1)^0 \times 1.0101010101 \times 2^7 = 10101010.101 = (170.625)_{10}$

- $(170.625)_{10} = (0001\ 0111\ 0000.0110\ 0010\ 0101)_{BCD}$

Q54

- Consider the following assembly program written for a computer with a two-addres instruction processor. If a portion of the memory is given as beside (V, W, X, Y, and Z are variables);

```
100    MOVE    R1, (200)
101    SUB     R1, (201)
102    MOVE    R2, (202)
103    ADD     R2, (204)
104    MPY     R1, R2
105    MOVE    (204), R1
```

Mem. Adress	Data
200	V = A4
201	W = A0
202	X = 72
203	Y = 22
204	Z = 41

Q54

- What function (in terms of the given variables) this program implements? (R1 and R2 are registers; MOVE: data movement instruction; SUB: subtraction instruction; MPY: multiplication instruction; ADD: addition instruction; DIV: division instruction)

- Assuming that V, W, X, Y, and Z are 2s complement signed numbers, what will be the values of *carry flag*, *sign flag*, *zero flag*, and *overflow flag* after the execution of instruction in memory location 103? Explain

A54

- Z = (V - W) (X + Y)

- 1

72 □ 0 1 1 1 0 0 1 0

41 □ 0 1 0 0 0 0 0 1

B3 □ 0 1 0 1 1 0 0 1 1

carry flag = 0 , sign flag = 1,

zero flag = 0, overflow flag = 1

Q55

- In the following MIPS code

```
lw      $t0,    32($s3)
add         $t0,    $s2,    $t0
sw      $t0,    48($s3)
```

  - How many times is instruction memory accessed?
  - How many times is data memory accessed?
  - How many times is register file accessed?
  - Which ones are read from the register file, which ones are write to the register file?

A55

- 3
- 2
- 7

- lw $t0, 32($s3) s0 is read from, t0 is written into

  add $t0, $s2, $t0 s2 and t0 are read from,
  t0 is written into

  sw $t0, 48($s3) s3 and t0 are read from

Q56

- The contents of memory and CPU registers of a computer system are given as following (the values are in hexadecimal).

Memory address	Memory content	Register Content	Register
1 0	**5 6 1**	**10**	**PC**
1 1	**1 6 0**		**AC**
1 2	**3 6 1**	**5 6 1**	**IR**
	.		
6 0	**0 0 3**		
6 1	**0 7 0**		
	.		
7 0	**0 0 5**		
7 1	**0 0 2**		

Q56

- What will be the content of the actual operand for the instruction being executed when the addressing mode is;

  - immediate addressing
  - direct addressing
  - indirect addressing
  - PC relative addressing

A56

- 061 (operand is adress)
- 070 (operand is the value in the address 61)
- 005 (operand is the value in the address 70 pointed by the address 61)
- 002 (operand is the value in the address calculated as 61+10 (content of PC))

Q57

- In an 8 bit microprocessor data transfer rate is given as 2 MByte per second.
- Find the clock frequency of the processor if a bus cycle takes 2 clock cycles.

A57

- If 2 MByte is transferred in 1 second, 1 Byte is transferred in **$1/(2\times10^{6}) = 0.5\times10^{-6}$ second** .

- If 1 byte requires 2 clock cycle to transfer, clock cycle is **$(0.5\times10^{-6})/2 = 0.25\times10^{-6}$** second.

- So, the clock frequency is **$1/(0.25\times10^{6}$ second) = 4 MHz**.

Q58

- Classify and briefly describe I/O methods that can be used in a computing system.

A58

- I/O can be controlled in four general ways.
  - Programmed I/O:
    - Reserves a register for each I/O device.
    - Each register is continually polled to detect data arrival.
  - Interrupt-Driven I/O:
    - Allows the CPU to do other things until I/O is requested.
  - Direct Memory Access (DMA):
    - Offloads I/O processing to a special-purpose chip that takes care of the details.
  - Channel I/O:
    - Uses dedicated I/O processors.

Q59

- Suppose a system has a virtual memory size of 64K and the system uses word addressing.

- If the page size is 1024 Word,
  - How many bits this virtual memory address requires?
  - Determine the number of virtual pages

A59

- virtual address space = 64K = $2^6 \times 2^{10} = 2^{16}$ Word
- page size = 1024 Word = 2^{10} Word

  — Because the virtual memory size is 2^{16} Word, the virtual memory address requires **16 bits**

  — Number of virtual pages = $2^{16} / 2^{10} = 2^{16-10} = 2^6 =$ **64**

Q60

- Assume that you have a computer with a maximum main memory size of 2 GByte (motherboard does not allow to upgrade the main memory).
- The program you want to run on this computer may require a larger main memory, resulting in a virtual memory that uses part of the hard drive.
  - However, you realized that the magnetic hard drive in this computer is too slow for your application.
- What can you do to increase the performance of this computer?
- Explain how and why your solution will increase the performance.

A60

- I would replace magnetic hard drive with a Solid State Drive (SSD).

- SSDs have the following advantages over HDDs:
  - High-performance input/output operations per second (IOPS),
  - less susceptible to physical shock and vibration longer lifespan,
  - lower power consumption,
  - lower access times and latency rates.

Q61

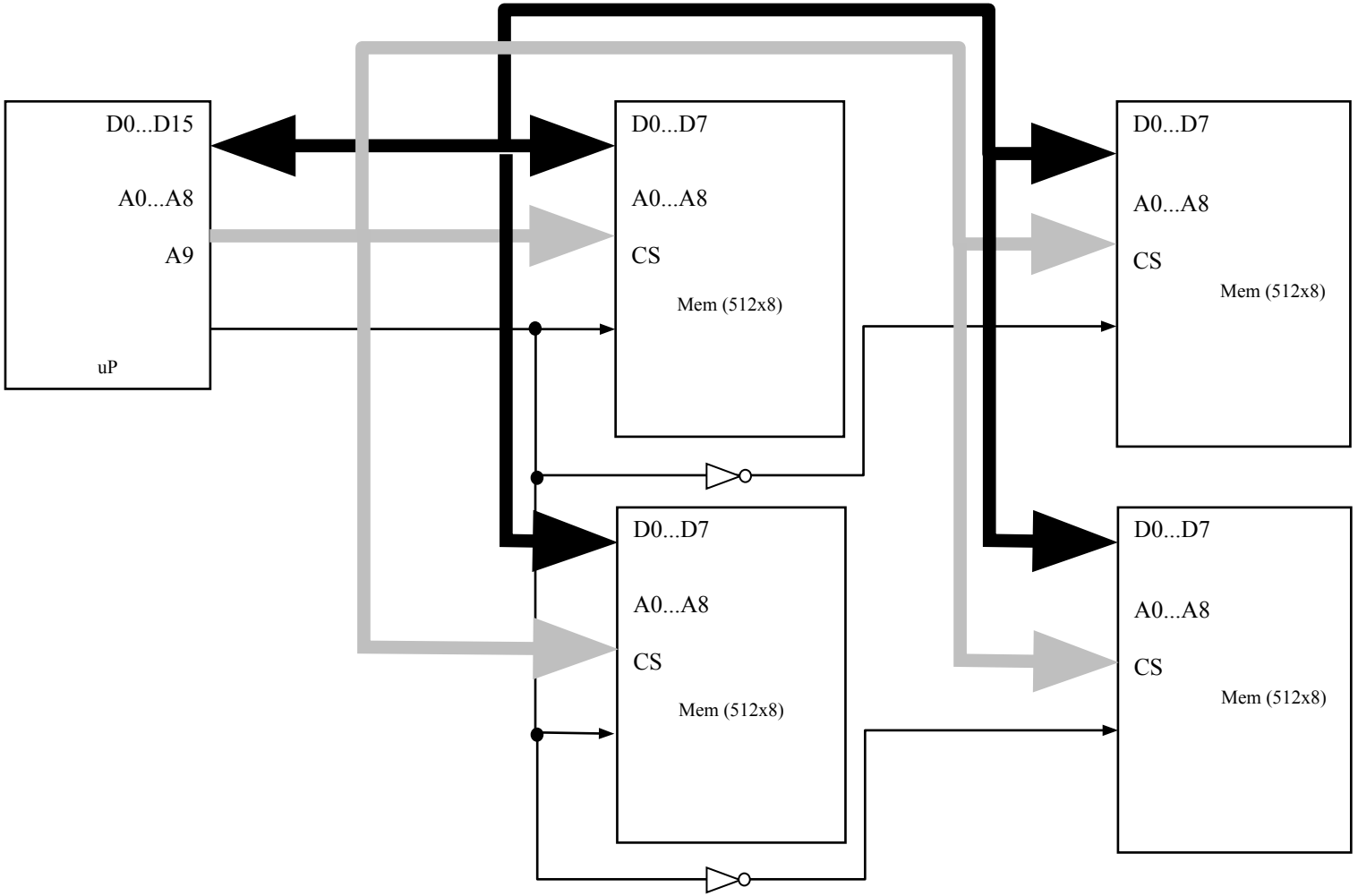
- A 16 bit microprocessor system has 10 address lines to address relevant memory locations.
  - Assuming that the data size is 1 byte, what is the address of the last memory location?
  - Assuming memory chips organized as 512x8 bits are used, how many memory chips we need to design the required memory).
  - Design the required memory system using the memory chips organized as 512x8 bits.

A61

- The last memory location
  = $2^{\#\text{of bits in address line}} - 1 = 2^{10} - 1 = 1023$

- Data bus size **16 bits**, memory size is **1024**. System memory organization is **1024x16**.
- If we restate this memory organization in terms of the available memory chip (512x8), we have {(2x512)x(2x8)} = **4**x(512x8).
  - So, We need **4** (512x8) memory chips to have a memory organized as 1024x16

A61


D0...D15
A0...A8
A9
uP
D0...D7
A0...A8
CS
Mem (512x8)
D0...D7
A0...A8
CS
Mem (512x8)
D0...D7
A0...A8
CS
Mem (512x8)
D0...D7
A0...A8
CS
Mem (512x8)

Q62

- Consider a processor which uses branch delay technique for optimization of pipelining. Each intruction has 2 or 3 stages depending on the instruction type (I: Instruction fetch, E: Execute D: Memory operation)

```
100 LOAD    R1, (500)
101 JUMP    105
102 ADD     R1, 1
103  AND    R1, (501)
```

  - Which instruction will be executed first after the processor encounters the instruction in line 101.
  - Draw the relevant pipelinening diagram for the given program.

A62

- First, instruction in line 100, which adds contents of memory location 500 to R1 will be executed.

- In line 101 there is a jump (unconditional branch) instruction.
  - Because the processor uses a branch delay technique, instruction in line 102 will be executed before processor jumps to line 105.

A62

	1	2	3	4	5	6
100 LOAD R1, (500)	I	E	D			
101 JUMP 105		I	E			
102 ADD R1, 1			I	E		
103 AND R1, (501)				I	E	D

Q63

- Why a cash and virtual memory are used in a computing system?
  - Briefly explain.

A63

- Cache memory enhances performance by providing faster memory access speed.
  - The purpose of cache memory to speed up accesses by storing recently used data closer to the CPU.
  - It is much smaller than main memory. Its access time is a fraction of that of main memory.
- Virtual memory enhances performance by providing greater memory capacity, without the expense of adding main memory.
  - Instead, a portion of a disk drive serves as an extension of main memory.

Q64

- Suppose we have 32-bit memory addresses, a byte-addressable memory, and a 512 KB cache with 32 bytes per block.

1. How many total lines are in the cache?
2. If the cache is direct-mapped, how many cache lines could a specific memory block be mapped to?
3. If the cache is direct-mapped, what would be the format (Tag-id, Line-id, Word-id) of the address? (Clearly indicate the number of bits in each)

Q64

4. If the cache is 2-way set associative (sets of size 2), how many cache lines could a specific memory block be mapped to?
5. If the cache is 2-way set associative (sets of size 2), how many sets would there be?
6. If the cache is 2-way set associative, what would be the format of the address?
7. If the cache is fully-associative, how many cache lines could a specific memory block be mapped to?
8. If the cache is fully-associative, what would be the format of the address?

A64

1. $\frac{512\ \text{KBytes}}{32\ \text{Bytes}} = \frac{2^{19}}{2^5} = 2^{14}\ \text{lines} = 2^4 \times 2^{10}\text{lines} = 16\ \text{K lines} = 16384\ \text{lines}$

2. **1** (a specific memory block could be mapped to only 1 line in the cache)

3. Each block contains 32 bytes. Therefore, 5 bits are needed to specify the *Word-id*.

   There are 2^{14} lines in the cache. Therefore, 14 bits are needed to specify the *Line-id*.

   Tag-id bits in this case is defined as *Main memory address* – (*Line-id bits* + *Word-id bits*). Therefore, 32 - (14 + 5) = 13 bits are needed to specify the *Tag-id*.

   The format of the address:

	Tag-id	Line-id	Word-id
Main memory address =	**13 bits**	**14 bits**	**5 bits**

A64

4. **2** (a specific memory block could be mapped to any 2 lines within 1 set in the cache)

5. $\frac{2^{14}\ \text{lines}}{2^{1}\ \text{lines}} = 2^{13}\ \text{sets} = 2^{3} \times 2^{10}\text{sets} = 8\ \text{K sets} = 80192\ \text{sets}$

6. Each block contains 32 bytes. Therefore, 5 bits are needed to specify the *Word-id*.
There are 2^{13} sets in the cache. Therefore, 13 bits are needed to specify the *Set-id*.

Tag-id bits in this case is defined as *Main memory address* – (*Set-id bits* + *Word-id bits*). Therefore, 32 - (13 + 5) = 14 bits are needed to specify the *Tag-id*.

	Tag-id	Set-id	Word-id
Main memory address =	**14 bits**	**13 bits**	**5 bits**

A64

7. 2^{14} (a specific memory block could be mapped to any cash line in the cache)

8. Each block contains 32 bytes. Therefore, 5 bits are needed to specify the *Word-id*.

Tag-id bits in this case is defined as *Main memory address – Word-id bits*. Therefore, 32 - 5 = 27 bits are needed to specify the *Tag-id*.

	Tag-id	**Word-id**
Main memory address =	**27 bits**	**5 bits**